WEST

Generate Collection Print

L7: Entry 5 of 26 File: USPT Sep 2, 2003

DOCUMENT-IDENTIFIER: US 6615223 B1
TITLE: Method and system for data replication

Abstract Text (1):
A method and mechanism for data replication is disclosed. One embodiment of the invention relates to an efficient and effective replication system using LDAP replication components. Another embodiment of the invention pertains to a schema and format independent method for data replication. Procedures for adding, deleting, and modifying replicated data, and for replicating conflict resolution are also disclosed. A further embodiment of the invention is directed to improved methods and mechanisms for adding and removing nodes from a replication system.

Brief Summary Text (3):
The invention relates to the replication of data in a database system.

Brief Summary Text (5):
Data replication is the process of maintaining multiple copies of a database object in a distributed database system. Performance improvements can be achieved when data replication is employed, since multiple access locations exist for the access and modification of the replicated data. For example, if multiple copies of a data object are maintained, an application can access the logically "closest" copy of the data object to improve access times and minimize network traffic. In addition, data replication provides greater fault tolerance in the event of a server failure, since the multiple copies of the data object effectively become online backup copies if a failure occurs.

Brief Summary Text (6):
In general, there are two types of propagation methodologies for data replication, referred to as "synchronous" and "asynchronous" replication. Synchronous replication is the propagation of changes to all replicas of a data object within the same transaction as the original change to a copy of that data object. For example, if a change is made to a table at a first replication site by a Transaction A, that change must be replicated to the corresponding tables at all other replication sites before the completion and commitment of Transaction A. Thus, synchronous replication can be considered real-time data replication. In contrast, asynchronous replication can be considered "store-and-forward" data replication, in which changes made to a copy of a data object can be propagated to other replicas of that data object at a later time. The change to the replicas of the modified data object does not have to be performed within the same transaction as the original calling transaction.

Brief Summary Text (7):
Synchronous replication typically results more overhead than asynchronous replication. More time is required to perform synchronous replication since a transaction cannot complete until all replication sites have finished performing the requested changes to the replicated data object. Moreover, a replication system that uses real-time propagation of replication data is highly dependent upon system and network availability, and mechanisms must be in place to ensure this availability. Thus, asynchronous replication is more generally favored for noncritical data replication activities. Synchronous replication is normally employed only when application requires that replicated sites remains continuously synchronized.

Brief Summary Text (8):
One approach to data replication involves the exact duplication of database schemas and data objects across all participating nodes in the replication environment. If this approach is used in a relational database system, each participating site in the replication environment has the same schema organization for the replicated database

tables and database objects that it maintains. If a change is made to one replica of a database table, that same change is propagated to all corresponding database tables to maintain the consistency of the replicated data. Since the same schema organization used the replicated data across all replication sites, the instructions used to implement the changes at all sites can be identical.

Brief Summary Text (9):
Generally, two types of change instructions have been employed in data replication systems. One approach involves the propagation of changed data values to each replication site. Under this approach, the new value for particular data objects are propagated to the remote replication sites. The corresponding data objects at the remote sites are thereafter replaced with the new values. A second approach is to use procedural replication. Under this approach, a database query language statement, e.g., a database statement in the Structured Query Language ("SQL"), is propagated instead of actual data values. The database statement is executed at the remote sites to replicate the changes to the data at the remote replication sites. Since all replication sites typically have the same schema organization and data objects, the same database statement can be used at both the original and remote sites to replicate any changes to the data.

Brief Summary Text (10):
A significant drawback to these replication approaches is that they cannot be employed in a heterogeneous environment in which the remote replication sites have different, and possibly unknown, schema organizations for the replicated data. For example, consider if information located in a single database table at a first replication site is stored within two separate tables at a second replication site. The approach of only propagating changed values for a data object to a remote replication site presents great difficulties, since the data object to be changed at the first replication site may not exist in the same form at the second replication site (e.g., because the data object exists as two separate data items at the second replication site). Using procedural replication results in similar problems. Since each replication site may have a different schema organization for its data, a different database statement may have to be specifically written to make the required changes at the remote sites. Moreover, if the schema organization of the remote site is unknown, it is impossible to properly formulate a database statement to replicate the intended changes at the remote site.

Brief Summary Text (11):
Another drawback to these approaches in which database schema and objects are exactly duplicated across the replication environment is that they require greater use of synchronous replication. If a schema change is made to a database table at one site, then that change must be synchronously propagated to all other sites. This is because the basic structure of the table itself is being changed. Any further changes to that database table without first synchronously changing the underlying schema for that table could result in conflicts to the data. Moreover, synchronous replication of the schema changes could require that the replication environment be quieced during the schema change, affecting the availability of the system.

Brief Summary Text (12):
One type of database application for which data replication is particularly useful is the replication of data for directory information systems. Directory information systems provide a framework for the storage and retrieval of information that are used to identify and locate the details of individuals and organizations, such as telephone numbers, postal addresses, and email addresses.

Brief Summary Text (18):
The present invention is directed to methods and mechanisms for data replication. According to an aspect of the invention, an efficient and effective replication system is disclosed using LDAP replication components. Another aspect of the invention pertains to a schema and format independent method and method for data replication. Yet another aspect of the invention relates to procedures for adding, deleting, and modifying replicated data and for replication conflict resolution. Another aspect of the invention relates to improved methods and mechanisms for adding and removing nodes from a replication system.

Drawing Description Text (4):
FIG. 1 depicts a system architecture for data replication according to an embodiment of the invention.

Detailed Description Text (2):
The present invention is directed to a method and mechanism for replication in a database system that does not depend upon the same schema or data organizations being maintained at each replication site. The present invention is particularly well suited for LDAP data replication. According to one aspect of the invention, any data changes at a first replication site are replicated to other replication sites using schema and system independent change records. The change records are created in a standard format that is usable by all other replication sites in the system. Once the change record has been propagated to each remote replication site, the change record is then utilized to implement database instructions that are appropriate for the specific schema and system parameters of the remote site.

Detailed Description Text (3):
FIG. 1 depicts a system architecture for performing data replication according to an embodiment of the invention. Note that FIG. 1 illustrates the invention with reference to two replication sites; however, the inventive principles described herein is equally applicable to systems having more than two replication sites.

Detailed Description Text (6):
For the purposes of illustration, assume that the system of FIG. 1 is used in a "peer-to-peer" or "multi-master" replication environment. In many peer-to-peer or multi-master replication environments, data changes made at a replication site are propagated to other replication sites, without the need for an overall "master" replication site. Thus, if a change request 12 at first replication site 2 is implemented to database data 6, that same change is replicated to the database data 56 at second replication site 52. Likewise, if a change request is made to second replication site 52 that is implemented to database data 56, that same change is replicated to the database data 6 at first replication site 2.

Detailed Description Text (26):
To perform data replication, a standard change record format is utilized to define LDAP data manipulation operations, in which the change record format is recognized and adhered to by each replication site. Change records are propagated to each replication site that describe the data changes made at the originating site. Regardless of the exact schema or data organization in place at each remote replication site, the LDAP server at each site comprises an LDAP engine that can interpret the standard format of the change records to replicate the changes to the local LDAP directory data. In this manner, peer-to-peer data replication can be performed in a heterogeneous environment in which local replication sites are not required to have knowledge of the exact schemas being employed by remote replication sites.

Detailed Description Text (32):
The embodiment of FIG. 3 also includes the use of a shadow log to propagate changes from one replication site to another. Change log entries from change log 314 are copied to a replication log 316 to be propagated to other replication sites. Replication log 316 is a shadow of change log 314, and its use prevents the need to bring down all LDAP databases when schema changes are propagated to the replication sites, such as the addition or deletion of LDAP databases from the replication environment. In essence, shadow logs are utilized to insulate the format of local replication logs from the actual mechanism used to propagate changes to other replication sites. In this manner, the internal schema formats of the replication sites are encapsulated by the shadow logs, such that schema changes can be made without downtime to the replication nodes.

Detailed Description Text (33):
A process runs at the LDAP directory site 302 to copy information from the change log 314 to the replication log 316. Either asynchronous or synchronous replication can be implemented using the invention. For asynchronous replication, the copying of entries from the change log 314 to the replication log 316 occurs either periodically, or upon certain specified trigger conditions. The change information is propagated and applied to remote LDAP sites in a queued "store-and-forward" process. For synchronous replication, the system constantly monitors the change log for the arrival of new entries. If a new entry is generated at the change log 314, the new entry is immediately copied to the replication log 316 for propagation to remote LDAP sites.

Detailed Description Text (34):
The change log information copied to the replication log 316 at the local LDAP directory site 302 is propagated to the replication log 320 at remote LDAP site 304. In the preferred embodiment, the mechanism used to replicate this information is the Advanced Symmetric Replication mechanism from the Oracle 8i database management system,

available from Oracle Corporation of Redwood Shores, Calif.

Detailed Description Text (35):
At the remote LDAP site 304, the change log entry in replication log 320 is directly sent to LDAP server 310 for processing. Alternatively, the change log entry in replication log 320 can be copied to change log 324 before being sent to LDAP server 310. A daemon process 322 initiates the application of the change log entry to the LDAP directory data 312 at LDAP site 304. If asynchronous replication is employed, the daemon process 322 wakes up periodically based upon defined intervals or upon specified trigger conditions to initiate the changes. If synchronous replication is employed, daemon process 322 actively monitors for any incoming change log information that has been propagated by a remote LDAP site. With synchronous replication, once the changes have been implemented, an acknowledgement is sent back to the propagating LDAP site.

Detailed Description Text (39):
FIG. 9 depicts the process flow of an embodiment of the invention to add a new LDAP site to an existing replication environment. The following describe the process actions of this process flow: 1. Stop the processes that propagate changes from change logs to replication logs tables at all sites (process action 902). 2. Redirect all LDAP functions from a master definition/configuration database (process action 904). In an embodiment of the invention, a master definition/configuration database is maintained to control configuration information regarding replication nodes, such as node identifiers, location, etc. Any of the replication nodes can be designated as the master definition/configuration site. 3. Suspend and quiesce the replication environment (process action 906). This ensures that all data presently at the replication logs are propagated to all sites by the replication mechanism. 4. Build a snapshot of the master definition/configuration database (process action 908). In an embodiment, building the snapshot comprises the performance of an online backup. A database log switch can be performed before the online backup. The master definition/configuration database can be triple-mirrored for quicker online backup. 5. Bring the master definition/configuration database back online (process action 910). 6. Resume all LDAP functions on master definition/configuration site (process action 912). 7. Add the new LDAP site to the replication environment, by adding the replication log table for the new site to the replicated environment and regenerating the replication support (914). At this point replication resumes between the LDAP sites. 8. Bring down the new LDAP directory site (process action 916). 9. Resume the jobs that copy information from change logs to replication logs (process action 918). Now all LDAP sites are fully available, except for the new LDAP database that is being added. 10. Bring up the LDAP new database (process action 920). This is performed by first bringing up the new database without the replication processes. The new database is then brought down and recreated using the backup of master definition/configuration database. Database administration changes are made for the new database (e.g., network names, database names, file names that may need to be changed, etc.). The Replication catalog tables are dropped into the new database and recreated. 11. At the new LDAP site, start replication processes as well as the processes that copy change information from the change log to the replication log (process action 922). 12. Start LDAP server and replication mechanism at the new LDAP site (process action 924).

Detailed Description Text (41):
FIG. 10 depicts the process flow of an embodiment of a process to remove an existing LDAP directory site from a replication environment. The following describe the process actions for this process flow: 1. Stop processes that propagate change information the change log and replication log at each LDAP directory site (process action 1002). 2. Quiesce the replication environment (process action 1004). 3. Drop the LDAP server from replication (process action 1006). 4. Resume replication activities at all other LDAP sites (process action 1008). 5. Start the process that were stopped in process action 1002 (process action 1010).

Detailed Description Text (86):
The term "computer-usable medium," as used herein, refers to any medium that provides information or is usable by the processor(s) 707. Such a medium may take many forms, including, but not limited to, non-volatile, volatile and transmission media. Non-volatile media, i.e., media that can retain information in the absence of power, includes the ROM 709. Volatile media, i.e., media that can not retain information in the absence of power, includes the main memory 708. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise the bus 706. Transmission media can also take the form of carrier waves; i.e., electromagnetic waves that can be modulated, as in frequency, amplitude or phase, to transmit information signals. Additionally, transmission media can take the form of acoustic or light waves,

- such as those generated during radio wave and infrared data communications.

Detailed Description Text (88):
Various forms of computer-usable media may be involved in providing one or more sequences of one or more instructions to the processor(s) 707 for execution. For example, the instructions may initially be provided on a magnetic disk of a remote computer (not shown). The remote computer may load the instructions into its dynamic memory and then transit them over a telephone line, using a modem. A modem local to the processing unit may receive the instructions on a telephone line and use an infrared transmitter to convert the instruction signals transmitted over the telephone line to corresponding infrared signals. An infrared detector (not shown) coupled to the bus 706 may receive the infrared signals and place the instructions therein on the bus 706. The bus 706 may carry the instructions to the main memory 708, from which the processor(s) 707 thereafter retrieves and executes the instructions. The instructions received by the main memory 708 may optionally be stored on the storage device 710, either before or after their execution by the processor(s) 707.

Detailed Description Text (89):
Each processing unit may also include a communication interface 714 coupled to the bus 706. The communication interface 714 provides two-way communication between the respective user stations 624-1, 624-2, 624-3, and 624-4 and the host computer 622. The communication interface 714 of a respective processing unit transmits and receives electrical, electromagnetic or optical signals that include data streams representing various types of information, including instructions, messages, and data.

Detailed Description Text (91):
A processing unit may transmit and receive messages, data, and instructions, including program, i.e., application, code, through its respective communication link 715 and communication interface 714. Received program code may be executed by the respective processor(s) 707 as it is received, and/or stored in the storage device 710, or other associated non-volatile media, for later execution. In this manner, a processing unit may receive messages, data and/or program code in the form of a carrier wave.

Current US Original Classification (1):
707/201

Current US Cross Reference Classification (1):
707/203

Current US Cross Reference Classification (2):
707/204